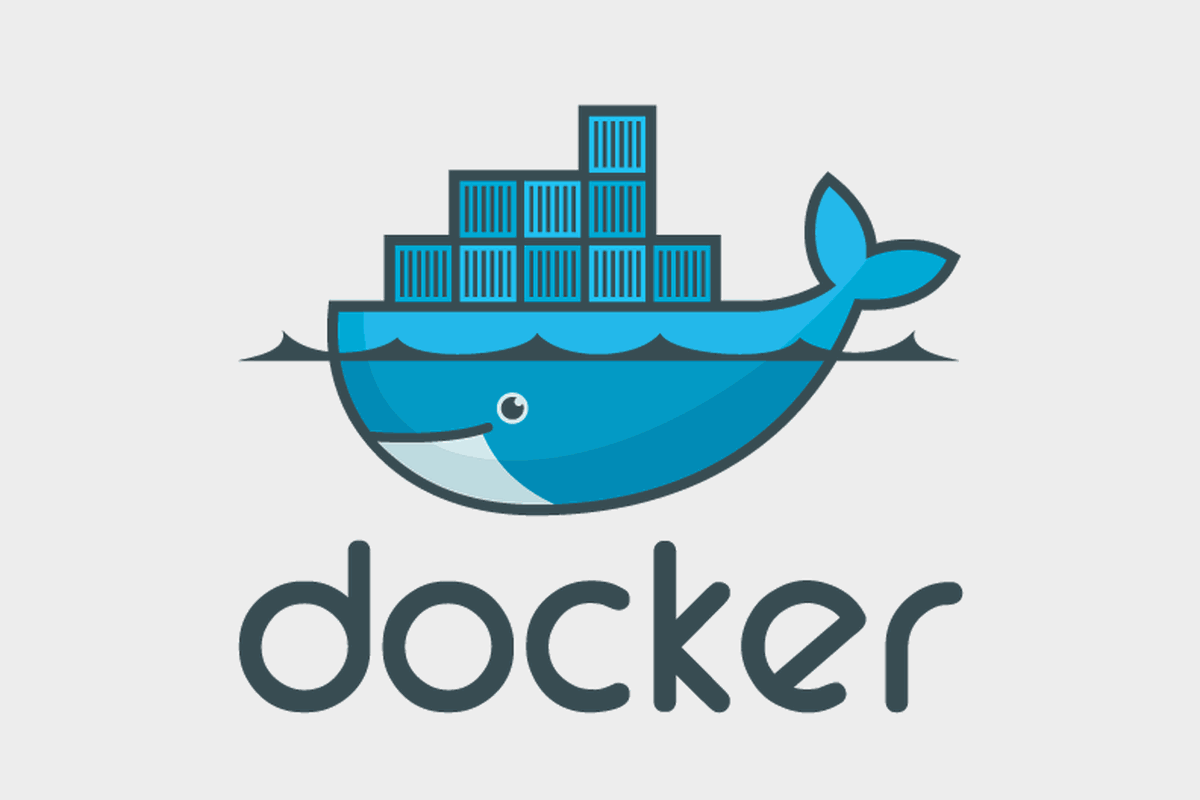
# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1 Docker và các thành phần của Docker

### 2.1.1 Docker là gì?

Docker là một dự án mã nguồn mở giúp tự động triển khai các ứng dụng Linux và Windows vào trong các container ảo hóa. Docker cung cấp một lớp trừu tượng và tự động ảo hóa dựa trên Linux.



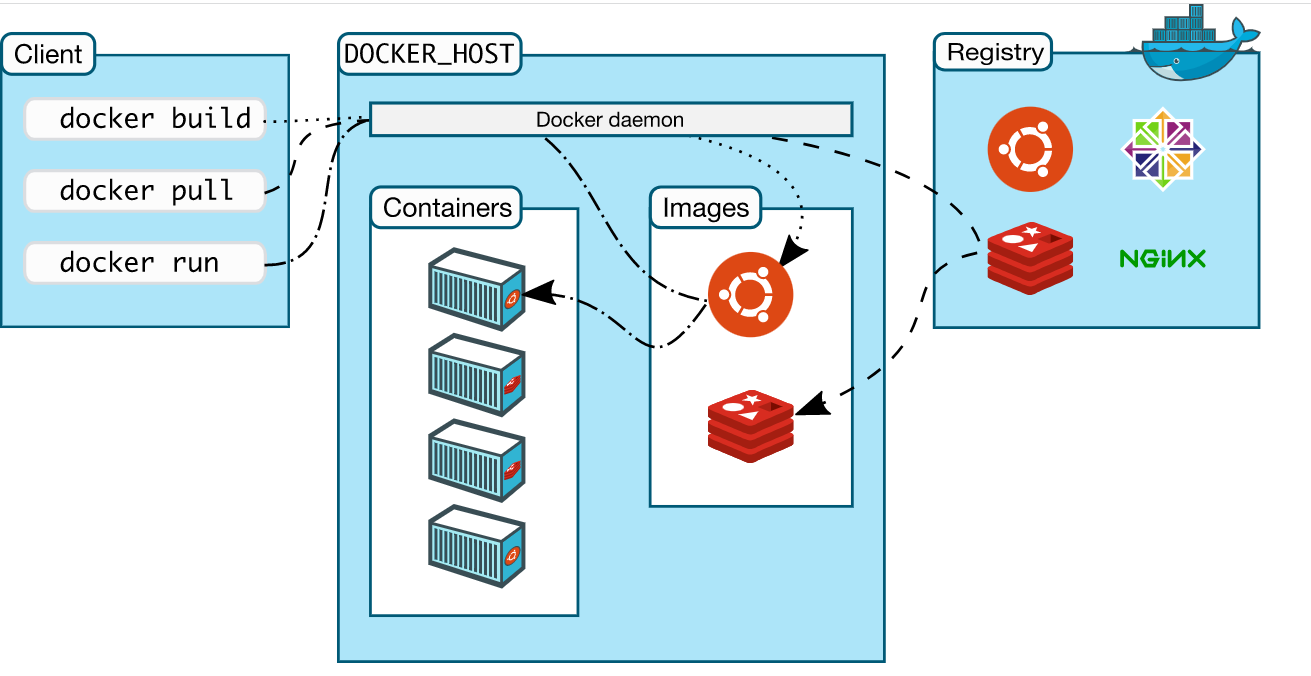
Docker là một công cụ giúp cho việc tạo ra và triển khai các container để phát triển, chạy ứng dụng được dễ dàng. Các container là môi trường, mà ở đó lập trình viên đưa vào các thành phần cần thiết để ứng dụng của họ chạy được, bằng cách đóng gói ứng dụng cùng với container như vậy, nó đảm bảo ứng dụng chạy được và giống nhau ở các máy khác nhau (Linux, Windows, MacOS, Windows Server …).

Docker có vẻ rất giống máy ảo (nhiều người từng tạo máy ảo với công cụ ảo hóa như Virtual Box, VMWare), nhưng có điểm khác với VM: thay vì tạo ra toàn bộ hệ thống (dù ảo hóa), Docker lại cho phép ứng dụng sử dụng nhân của hệ điều hành đang chạy Docker để chạy ứng dụng bằng cách bổ sung thêm các thành phần còn thiếu cung cấp bởi container. Cách này làm tăng hiệu xuất và giảm kích thước ứng dụng.

Hiểu đơn giản hơn thì Docker cung cấp cho người sử dụng những service và những công cụ cần thiết để giúp người sử dụng có thể chạy các chương trình và đóng gói chúng ở trong tất cả môi trường khác nhau một cách nhanh chóng và đơn giản nhất.

Tóm lại, Docker là một công cụ giúp deploy dự án phần mềm một cách nhanh chóng trong các container. Các môi trường liên quan cần thiết để phần mềm hoạt động đều được Docker cung cấp đầy đủ. Triển khai dự án phần mềm một cách nhanh chóng và dễ dàng. Có rất nhiều lợi ích đáng để sử dụng Docker.

### 2.1.2 Docker Engine

Docker Engine (Docker Daemon) là một thành phần quan trọng của Docker. Nó chính là thành phần chính giúp vận hành Docker. Có thể hiểu đây chính là cốt lõi của Docker. 

Docker Engine được cài trên các máy chủ sử dụng Docker (Docker Host), nó chứa các đối tượng là thành phần con của docker (Container, Image) quản lý và vận hành chúng.

Docker Engine là công cụ Client - Server hỗ trợ công nghệ container để xử lý các nhiệm vụ và quy trình công việc liên quan đến việc xây dựng các ứng dụng dựa trên vùng chứa (container). Engine tạo ra một quy trình daemon phía máy chủ lưu trữ images, containers, networks và storage volumes. Daemon cũng cung cấp giao diện dòng lệnh phía máy khách (CLI) cho phép người dùng tương tác với daemon thông qua giao diện lập trình ứng dụng Docker.

### 2.1.3 Docker Image

Docker Image là một file có chứa tất cả các source code, libraries, dependencies, tools và các files khác cần thiết để ứng dụng có thể hoạt động. Docker image

Image chỉ là các mẫu để từ đó build các Container tương ứng. Docker Image không cho phép deploy trực tiếp phần mềm lên chúng.

Image là là thành phần để đóng gói ứng dụng và các thành phần mà ứng dụng phụ thuộc để chạy. Image có thể lưu local hoặc lưu Registry

Hiểu theo cách đơn giản nhất, Docker Image giống như hình ảnh mô phòng của một môi trường thực thi với mục đích nhất định. Chúng chứa các source code, các thư viện để phục vụ cho việc chạy một phần mềm.

Ví dụ dễ hiểu về các Image: Image npm dùng chạy nodeJs, Image Java JDK dùng để chạy Java, Python . . .

### 2.1.4 Docker Container

Nếu như coi Docker Image là một class thì Docker Container là một instance của image. Container cung cấp môi trường thực cho ứng dụng hoạt động. Ở đây, người dùng có thể chạy các ứng dụng phần mềm.

Container nắm giữ toàn bộ các package cần thiết để khởi động và chạy ứng dụng. Các Container sẽ sử dụng chung tài nguyên của hệ thống nên rất nhẹ và giúp các thao tác kết nối, tương tác diễn ra nhanh chóng, tiện lợi hơn.

Container có công dụng giống hệt với tên của nó. Container như một “vùng chứa”. Soure code phần mềm của bạn khi deploy sẽ chạy trực tiếp trong container này. Bạn có thể hình dung khi đó container như một server riêng, máy ảo riêng để chạy phần mềm.

Ưu điểm của Container là rất nhẹ và dễ dàng tạo, xóa. Đây chính là một trong những ưu điểm lớn nhất của docker container so với các công nghệ máy ảo khác.

### 2.1.5 Dockerfile và Docker Compose

*a, Docker file*

Dockerfile là thành phần của hệ sinh thái Docker, các Docker file khi build sẽ tạo thành Docker Image. Chính nhờ Dockerfile mà lập trình viên có thể tự xây dựng các Image, Container cần thiết cho project của bạn.

Dockerfile là một file dạng text không có đuôi, Dockerfile có cấu trúc theo quy chuẩn nhất định của Docker. Nội dung gồm tập hợp các câu lệnh cũng như phụ thuộc cần thiết để tạo ra Docker Image.

Ví dụ project của bạn cần một Image có cài cả hai môi trường NodeJS và Python thì trong Dockerfile bạn có gọi chúng vào là được. Tất cả việc còn lại Docker sẽ giúp bạn.

Dockerfile sinh ra giúp việc deploy thực nhanh hơn rất nhiều, khi triển khai phần mềm chỉ cần có dockerfile. Điều này giúp tạo Docker Image và Container nhanh hơn rất nhiều.

*b, Docker Compose*

Docker Compose là công cụ dùng để định nghĩa và run multi-container cho Docker application. Docker Compose, chúng ta sử dụng một file YAML để thiết lập các service cần thiết cho chương trình. Cuối cùng, với một câu lệnh, chúng ta sẽ tạo và khởi động tất cả các service từ các thiết lập đó.

Khi phát triển một chương trình, việc chạy một chương trình trong một môi trường cô lập và tương tác là rất cần thiết. Compose cho phép thiết lập và chạy tất cả các service cần thiết cho chương trình. Chỉ với một câu lệnh docker-compose up, các service đó sẽ được chạy với các container tương ứng

### 2.1.6 Docker Hub

Docker Hub là Registry, Container Image Library chỉnh thức của Docker, là một dịch vụ do Docker cung cấp, cho phép tìm kiếm và chia sẻ các Container Image.

Các tính năng chính của Docker Hub là:

* **Repositories**: Push và pull container images.
* **Teams & Organizations**: Quản lý quyền truy cập vào private repositories của container images.
* **Official Images**: Pull sử dụng container images chất lượng cao của Docker.
* **Publisher Images**: Pull và sử dụng container images được cung cấp bởi vendors khác.
* **Builds**: Tự động tạo container images từ GitHub và Bitbucket. Push chúng lên Docker Hub.
* **Webhooks**: Kích hoạt các actions sau khi push thành công một repository lên Docker Hub với các dịch vụ khác.